

~~FILED UNDER SEAL PURSUANT TO PROTECTIVE ORDER~~

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

SAS INSTITUTE INC.,

Plaintiff,

v.

WORLD PROGRAMMING LIMITED,
LUMINEX SOFTWARE, INC., YUM!
BRANDS, INC., PIZZA HUT, INC.,
SHAW INDUSTRIES GROUP, INC.,

Defendants.

Civil Action No. 2:18-CV-00295-JRG

Jury Trial Demanded

~~FILED UNDER SEAL~~

PUBLIC VERSION

**WORLD PROGRAMMING LTD.'S RESPONSE BRIEF
FOR COPYRIGHTABILITY HEARING**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	SASI OFFERS NO EVIDENCE OF FILTRATION.....	1
III.	FACTUAL BACKGROUND.....	3
	A. Background Related to the SAS System Source Code, the Asserted Work.....	3
	B. Background to SAS Institute’s Constantly Shifting Copyright Theories to Gain a Century-Long Monopoly on a Language.	4
IV.	LEGAL FRAMEWORK	6
V.	THE COURT SHOULD FIND THAT SASI HAS FAILED TO ESTABLISH ANY CORE PROTECTABLE EXPRESSION IN THE ASSERTED WORKS	7
	A. The Procedural Framework Underlying the Court’s Analysis.....	7
	1. The Court Must Conduct an Abstraction-Filtration Test Prior to Presenting Substantial Similarity to the Jury.	7
	2. The Burden to Complete Valid Abstraction and Filtration Is SASI’s.....	9
	3. Any Registration Presumption Is Insufficient to Carry SASI’s Burden.	11
	4. The Court May Resolve All Disputed Facts in Connection with Its Abstraction and Filtration Analysis.	12
	B. On This Record, There Is No Core Protectable Expression.....	13
	1. The Court Can Find No Core Protectable Expression Because SASI Failed to Meet Its <i>Altai</i> Burden.	14
	2. There Is No Core Protectable Expression Related to the Asserted Works’ “Collection of Input Formats” or “Naming” of SAS Language syntax.....	16
	3. There Is No Core Protectable Expression Related to the Asserted Work’s Relevant “Output Designs.”	21
	4. <i>Google v. Oracle</i> Does Not Save SASI From Its Failure to Prove Any Protectable Expression.	25
	C. A Proper <i>Altai</i> Analysis Would Proceed Like Dr. Jones’s.....	26
VI.	THE COURT SHOULD DENY SAS INSTITUTE’S REQUEST FOR DELAY	30
VII.	CONCLUSION.....	30

TABLE OF AUTHORITIES

Cases

<i>Cisco Sys., Inc. v. Arista Networks, Inc.</i> 5:14-cv-05344-BLF, Doc. 633 (N.D. Cal. Nov. 10, 2016).....	17, 21
<i>Cisco Sys., Inc. v. Huawei Techs., Co.</i> 266 F. Supp. 2d 551 (E.D. Tex. 2003).....	2
<i>Compaq Computer Corp. v. Ergonomics, Inc.</i> 137 F. Supp. 2d 768 (S.D. Tex. 2001).....	13
<i>Compulife Software Inc. v. Newman</i> 959 F.3d 1288 (11th Cir. 2020)	9, 10, 11
<i>Computer Associates Intern., Inc. v. Altai, Inc.</i> 982 F.2d 693 (2d Cir. 1992)	8
<i>Computer Mgmt. Assistance Co. v. Robert F. DeCastro, Inc.</i> 220 F.3d 396 (5th Cir. 2000)	2, 6, 16, 28
<i>Design Data Corp. v. Unigate Enter., Inc.</i> 847 F.3d 1169 (9th Cir. 2017)	23
<i>Digital Drilling Data Sys. LLC v. Petrolink Servs. Inc.</i> No. 4:15-CV-02172, 2018 WL 2267139 (S.D. Tex. May 16, 2018).....	23, 24
<i>Eng'g Dynamics, Inc. v. Structural Software, Inc.</i> 26 F.3d 1335 (5th Cir. 1994)	passim
<i>Eng'g Dynamics, Inc. v. Structural Software, Inc.</i> 46 F.3d 408 (5th Cir. 1995)	20, 21, 30
<i>Feist Publications, Inc. v. Rural Tel. Serv. Co.</i> 499 U.S 340 (1991).....	passim
<i>Gates Rubber Co. v. Bando Chem. Indus., Ltd.</i> 9 F.3d 823 (10th Cir. 1993)	passim
<i>General Universal Systems, Inc. v. Lee</i> 379 F.3d 131 (5th Cir. 2004)	passim
<i>Harbor Software, Inc. v. Applied Sys., Inc.</i> 925 F. Supp. 1042 (S.D.N.Y. 1996)	12

<i>Hutchins v. Zoll Med. Corp.</i> 492 F.3d 1377 (Fed. Cir. 2007)	7
<i>Jane Envy, LLC v. Infinite Classic Inc.</i> SA:14-CV-065-DAE, 2016 WL 797612 (W.D. Tex. Feb. 26, 2016).....	13
<i>Kepner-Tregoe, Inc. v. Leadership Software, Inc.</i> 12 F.3d 527 (5th Cir. 1994)	6, 11, 12
<i>Lakedreams v. Taylor</i> 932 F.2d 1103 (5th Cir. 1991)	11
<i>Lexmark Intern., Inc. v. Static Control Components, Inc.</i> 387 F.3d 522 (6th Cir. 2004)	11
<i>Macro Niche Software, Inc. v. 4 Imaging Sols., L.L.C.</i> No. CV H-12-2293, 2013 WL 12140417 (S.D. Tex. Dec. 18, 2013).....	2
<i>Mason v. Montgomery Data, Inc.</i> 967 F.2d 135 (5th Cir. 1992)	11
<i>M-I LLC v. Q'Max Solutions Inc. et al.</i> No. H-18-1099, 2020 WL 4549210 (S.D. Tex. Aug. 6, 2020).....	2, 16
<i>MiTek Holdings, Inc. v. Arce Eng'g Co.</i> 89 F.3d 1548 (11th Cir. 1996)	10
<i>Norma Ribbon & Trimming, Inc. v. Little</i> 51 F.3d 45 (5th Cir. 1995)	11
<i>Oracle Am., Inc. v. Google Inc.</i> 810 F. Supp. 2d 1002 (N.D. Cal. 2011).....	11
<i>Oracle Am., Inc. v. Google Inc.</i> 750 F.3d 1339 (Fed. Cir. 2014)	passim
<i>Pepper v. Int'l Gaming Sys., LLC</i> 312 F. Supp. 2d 853 (N.D. Miss. 2004).....	15
<i>R. Ready Productions, Inc. v. Cantrell</i> 85 F. Supp. 2d 672 (S.D. Tex. 2000).....	10
<i>Rearden LLC v. Walt Disney Co.</i> 293 F. Supp. 3d 963 (N.D. Cal. 2018)	23, 24, 25

S & H Computer Sys., Inc. v. SAS Inst., Inc.
568 F. Supp. 416 (M.D. Tenn. 1983)..... 4

SAS Inst. Inc. v. World Prog'g Ltd.
64 F. Supp. 3d 755 (E.D.N.C. 2014) 5, 18

Taylor v. IBM
54 F. App'x 794 (5th Cir. 2002) 6, 28

Torah Soft Ltd. v. Drosnin
136 F. Supp. 2d 276 (S.D.N.Y. 2001) 2, 23, 24

Statutes

17 U.S.C. § 102(a)	6, 22, 23
17 U.S.C. § 102(b)	6
17 U.S.C. § 410(a)	12
17 U.S.C. § 410(c)	12
17 U.S.C.A. § 101	7
37 C.F.R. § 202.1(a) (2001).....	6

Rules

Fed. R. Civ. P. 26(a)(2).....	15
Fed. R. Civ. P. 30(b)(6).....	23
Fed. R. Evid. 301	12

Other Authorities

3 Nimmer, § 13.03[F]	8
C.J. Sprigman & S.F. Hedrick, <i>The Filtration Problem in Copyright's "Substantial Similarity" Infringement Test</i> 23 LEWIS & CLARK L. REV. 571 (2019).....	13
SCOTT ON INFO. TECH. L. § 2.51 (3d Ed. 2020-2 Supplement).....	9
U.S. Copyright Office Compendium of U.S. Copyright Office Practices § 721.10(A) (3d ed.) (2017).....	22

I. INTRODUCTION

As SAS Institute Inc. (“SASI”) acknowledges, this is a copyright case over “non-literal” computer program elements that SASI calls its “input formats” and “output designs.” This means that SASI is not accusing World Programming Ltd. (“WPL”) of copying SASI’s literal copyrighted work—the SAS source code—but rather, some “non-literal” aspect of that code. SASI is forced to assert “non-literal” elements because the source code implementation of the accused work WPS is nothing like that of the SAS source code. The similarity is with respect to what each program *does*, not how each looks. The former is not the province of copyrights.

In copyright cases of literal works, the asserted copyright is apparent from the work itself (*e.g.*, reading the book); the same is not true for “non-literal” program elements. If a plaintiff wishes to extend its protection to encompass “non-literal” aspects, it has a burden to define that expression. That definition requires a valid *Altai* abstraction-filtration-comparison (“AFC”) analysis. The AFC is necessary to determine the extent to which the claimed “non-literal” expression is eligible for protection at all, and if so, whether it contains any protectable expression after filtering. Filtration is particularly necessary in computer works where “as one moves away from the literal code to more general levels of a program, it becomes more difficult to distinguish between unprotectible ideas, processes, methods or functions, on one hand, and copyrightable expression on the other.” *Eng’g Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1341 (5th Cir. 1994), *opinion supplemented on denial of reh’g*, 46 F.3d 408 (5th Cir. 1995).

II. SASI OFFERS NO EVIDENCE OF FILTRATION.

The Court ordered the parties to “present evidence in support of the abstraction and filtration steps of the abstraction-filtration-comparison (‘AFC’) test[.]”. Dkt. 425 at 2. SASI did not comply with the Court’s order. Specifically, SASI presents no evidence of filtration for its asserted non-literal elements—input formats, output designs, and naming of individual elements.

This failure results in an invalid AFC test because a valid AFC test requires filtering out noncopyrightable elements “from each particular level of a program.” *Eng’g Dynamics*, 26 F.3d at 1344. Yet for its three asserted levels, SASI filters nothing. SASI refused to submit a valid AFC test throughout this case. Its failure to do so now—in the face of the Court’s order to submit filtration evidence—is fatal to SASI’s case. SASI cannot sidestep the filtration requirement by asserting that filtration is not required. By acting as if “all elements of [the asserted work] are copyrightable,” it is “essentially asserting the filtration step does not have to be performed, which contradicts Fifth Circuit precedent.” *Macro Niche Software, Inc. v. 4 Imaging Sols., L.L.C.*, No. CV H-12-2293, 2013 WL 12140417, at *5 (S.D. Tex. Dec. 18, 2013).

SASI’s refusal stems from its belief that copyrightability in this case should be judged solely by the “minimal degree of creativity” framework expressed in *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340 (1991). But while “minimal creativity” is an initial step in identifying what might be copyrightable, it is not the last. The Fifth Circuit requires filtration to identify non-literal program elements that are actually copyrightable. *See Computer Mgmt. Assistance Co. v. Robert F. DeCastro, Inc.*, 220 F.3d 396, 400 (5th Cir. 2000).¹ Were filtration instead optional, this “minimal degree of creativity” standard would become a camel’s nose that could confer copyrightability on the noncopyrightable.

Seeking to do exactly that, SASI has regularly conflated what *might* be copyrightable to what actually is. *See* Dkt. 441 (Opening Br.) at 17 (invoking *Feist* principles of “creativity and

¹ When the asserted infringement is over “non-literal” extensions of a registered computer program, the “extremely low” threshold of *Feist* is supplemented by a rigorous AFC analysis. *See M-I LLC v. Q’Max Solutions Inc. et al.*, No. H-18-1099, 2020 WL 4549210 (S.D. Tex. Aug. 6, 2020) (“M-I argues that the court should instead hold that the nonliteral elements of its Copyrighted Works are protectable based on tests stated in *Torah Soft* and *Feist*. M-I’s argument lacks merit.”); *Cisco Sys., Inc. v. Huawei Techs., Co.*, 266 F. Supp. 2d 551, 554 (E.D. Tex. 2003) (noting Fifth Circuit assesses non-literal elements using abstraction/filtration/comparison method).

expressiveness.”); Dkt. 264 at 30 (“the collection of input formats and output designs to pass the ‘extremely low’ threshold required for copyrightability.”); Dkt. 328 at 4-5 (creativity “[a]ll that is necessary for copyrightability”). Likewise, SASI’s expert, Dr. Storer failed to perform filtration under the false assumption that a modicum of creativity negates the need to filter elements that cannot be copyrighted. SASI’s argument is untenable. And in seeking protection for every choice made in creating the SAS System, its position looks suspiciously like the “sweat of the brow” doctrine that the *Feist* Court rejected. *Feist*, 499 U.S. at 352. Here, allowing SASI to submit substantial similarity to a jury without filtration would be reversible error. *See Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 843 (10th Cir. 1993) (finding district court erred by failing to filter before addressing the ultimate question of infringement).

SASI has not met its burden to offer a valid AFC test. And WPL’s valid AFC analysis demonstrates that no core protectable expression exists within SASI’s Asserted Works. Either way, the allegedly copyrighted “inputs” and “outputs” are not SASI’s protectable expression.

III. FACTUAL BACKGROUND

A. Background Related to the SAS System Source Code, the Asserted Work.

Although SASI’s copyright assertion in this case has been a moving target, it alleges infringement of a statistical analysis computer program it refers to in briefing as the “SAS System.” SASI uses this moniker to ensnare dozens of separate copyright registrations across some, but not all, versions of its computer program released over time (“the Asserted Works”). These computer programs read, parse, and execute user-written programs for manipulating and analyzing users’ data sets. WPL has never had access to SAS source code.

The Asserted Works are not wholly original to SASI. To the contrary, SASI’s predecessors created the first versions of the SAS software under contract with the United States Government, and at least one of these works, SAS 76, has been conclusively determined to

reside in the public domain. *See S & H Computer Sys., Inc. v. SAS Inst., Inc.*, 568 F. Supp. 416, 419 (M.D. Tenn. 1983); Ex. A (Goodnight Dep. Tr.) at 63:13-15. While the *source code* of SAS 76 has been re-written since then, the Asserted Works still use the same non-literal elements from this public domain version, spanning from the core program structure down to identically named language elements. It is also undisputed that the Asserted Works borrow from more than just the public domain SAS 76: SASI also “copied” third-party competitors. For example, SASI and its predecessors took from other statistical analysis options, like PL/I and SPSS, in developing SAS 76 and later versions—going so far as to name its “PROCS”—“the heart of the SAS System,” Opening Br. at 14, because PL/I’s software did so.

In connection with creating the computer program, SASI’s predecessors also developed a computer programming language, the SAS Language. While the language has evolved over time (as all languages do), the syntax, grammar, and many of its “words” were all contained within the public domain SAS 76 version.

B. Background to SAS Institute’s Constantly Shifting Copyright Theories to Gain a Century-Long Monopoly on a Language.

For eleven years, SASI has chased WPL around the world with claims of copyright infringement. SASI’s objective is to monopolize the *functionality* for compiling SAS Language programs through its *copyrights*. Given the incompatibility between the two, SASI has had to brand and rebrand its copyright allegations to mimic other cases. Originally, SASI contended that WPL’s program WPS infringed SASI’s copyrights because of its ability to understand the SAS Language. Indeed, in the U.K. litigation, SASI submitted that “WPL had reproduced the SAS Language and that this was an infringement of the copyright in the SAS Language.” Ex. B (*SAS Institute Inc. v. World Programming Ltd*, [2013] EWHC January 25 2013 Order) at ¶ 77. The court held that the SAS Language would not be entitled to copyright protection. *Id.* at ¶ 78.

In the subsequent North Carolina case, SASI took the *same* elements but rebranded them: now, they were “inputs.” *See SAS Inst. Inc. v. World Programming Ltd.*, 64 F. Supp. 3d 755, 778 (E.D.N.C. 2014), *aff’d in part, vacated in part*, 874 F.3d 370 (4th Cir. 2017). Hoping to gain traction with a then-recent *Oracle v. Google* case, SASI even went so far as to characterize the SAS Language as SASI’s “declaring code.” *Id.*, Dkt. 243 (SASI Opp’n to MSJ) at 14. The North Carolina court rejected these new theories as word play, noting that SASI was “characteriz[ing] certain procedures (referred to as ‘PROCS’) and their accompanying PROC statements invoking these procedures as ‘inputs’ into the SAS System . . . rather than as elements of the SAS Language,” before granting summary judgment of no copyright infringement. 64 F. Supp. 3d at 778. The Court reasoned that, “by asking the court to find that [WPL’s] software infringes its copyright through its processing of elements the SAS Language,” SASI sought “to copyright the idea of a program which interprets and compiles the SAS Language.” *Id.* at 776.

In this third case with the same elements of the Asserted Works, SASI again eschews its “SAS Language” label and rebrands its assertions yet again. Now, hoping to mirror *Engineering Dynamics*, SASI claims that the SAS Language is a “collection of input formats.”² *See Eng’g Dynamics*, 26 F.3d at 1342. And the SAS Language is no longer SASI’s “declaring code”; the PROCs are now akin to “the Oracle API packages,” an entirely different assertion. Dkt. 441 at 14. But a decade’s worth of surface-level rebranding does nothing to transform uncopyrightable material into protectable expression. The core of SASI’s allegations remains the same: SASI is trying to extend its software copyrights to cover the idea and functionality of the SAS Language.

² Each of the elements identified by SASI as part of the Asserted Work’s “input formats” are elements of the SAS Language syntax. *See* Ex. D (2/11/14 30(b)(6) Depo.) at 8-9 (syntax) 13-14 (formats), 15 (procs), 15-16 (global statements) 16-17 (informats), 23-24 (statements), 24-25 (options), access engines (42). *See also* 64 F. Supp. 3d at 775-76.

This contention has been rejected on two separate occasions by two different legal systems.

IV. LEGAL FRAMEWORK

“Copyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression.” 17 U.S.C. § 102(a). “Copyright protection may extend only to those components of a work that are original to the author.” *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 348 (1991) (citations omitted). In addition to protecting only original expressions, a copyright owner also may not protect any “idea, procedure, process, system, method of operation, concept, principle, or discovery.” 17 U.S.C. § 102(b).

The following other doctrines also govern various elements that are not protectable:

- **merger** – “When an idea can be expressed in very few ways, copyright law does not protect that expression, because doing so would confer a *de facto* monopoly over the idea. In such cases idea and expression are said to be merged.” *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 533 (5th Cir. 1994).
- **scènes à faire** – expression that are standard, stock, or common to a particular subject matter, elements serving functional purposes, or elements dictated by external factors. *Eng’g Dynamics*, 26 F.3d at 1344 (citation omitted); *Computer Mgmt. Assistance Co.*, 220 F.3d 396, 401 (5th Cir. 2000).
- **other unprotectable material:** (1) short words or phrases, *Taylor v. IBM*, 54 F. App’x 794 (5th Cir. 2002) (citing 37 C.F.R. § 202.1(a) (2001)); (2) facts, *Eng’g Dynamics, Inc.*, 26 F.3d at 1344; and (3) public domain material, *id.*

“[C]omputer programs are in principle entitled to copyright protection.” *Eng’g Dynamics*, 26 F.3d at 1341 (footnote omitted).³ “This protection extends not only to the ‘literal’ elements of computer software—the source code and object code—but also to a program’s nonliteral elements, including its structure, sequence, organization, user interface, screen displays, and menu structures.” *Gen. Universal Sys.*, 379 F.3d at 142 (footnotes omitted).⁴ But protection does

³ “A ‘computer program’ is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.” 17 U.S.C.A. § 101.

⁴ “Source code is the programming language readable by human programmers; object code is the

not extend to the methods that are performed with program guidance” *Hutchins v. Zoll Med. Corp.*, 492 F.3d 1377, 1383 (Fed. Cir. 2007). In other words, “copyright does not protect the technologic process independent of the program that carries it out; that is, the copyright covers the way the process is described in the written or electronic form of the computer program, but does not cover the process independent of the copyrighted program.” *See id.* For highly functional aspects of computer programs, like computer user interfaces or highly standardized technical information, copyright protection, if any, is thin. *Eng’g Dynamics*, 26 F.3d at 1348.

V. THE COURT SHOULD FIND THAT SASI HAS FAILED TO ESTABLISH ANY CORE PROTECTABLE EXPRESSION IN THE ASSERTED WORKS

A. The Procedural Framework Underlying the Court's Analysis.

1. The Court Must Conduct an Abstraction-Filtration Test Prior to Presenting Substantial Similarity to the Jury.

“To assess a claim of software infringement, [the Fifth Circuit has] generally endorsed the ‘abstraction-filtration-comparison’ test first outlined by the Second Circuit in *Altai* and refined by the Tenth Circuit in *Gates Rubber Co. v. Bando Chemical Industries, Ltd.*” *Gen. Universal Sys.*, 379 F.3d at 142 (footnote omitted). The test is summarized as follows:

First, in order to provide a framework for analysis, we conclude that a court should dissect the program according to its varying levels of generality as provided in the abstractions test. Second, poised with this framework, the court should examine each level of abstraction in order to filter out those elements of the program which are unprotectable. Filtration should eliminate from comparison the unprotectable elements of ideas, processes, facts, public domain information, merger material, scenes a faire material, and other unprotectable elements suggested by the particular facts of the program under examination. Third, the court should then compare the remaining protectable elements with the allegedly infringing program to determine whether the defendants have misappropriated substantial elements of the plaintiff's program.

Eng'g Dynamics, 26 F.3d at 1342-43.

binary expression that controls the computer hardware.” *Eng’g Dynamics*, 26 F.3d at 1341, n.7.

In the abstraction step, “in a manner that resembles reverse engineering on a theoretical plane, a court should dissect the allegedly copied program’s structure and isolate each level of abstraction contained within it. This process begins with the code and ends with an articulation of the program’s ultimate function. Along the way, it is necessary essentially to retrace and map each of the designer’s steps—in the opposite order in which they were taken during the program’s creation.” *Computer Associates Intern., Inc. v. Altai, Inc.*, 982 F.2d 693, 707 (2d Cir. 1992) (citation omitted). This is intended to “help a court separate ideas [and processes] from expression and eliminate from the substantial similarity analysis those portions of the work that are not eligible for copyright protection.” *Eng’g Dynamics*, 26 F.3d at 1343 (quoting 3 Nimmer, § 13.03[F] at 13–102.17). Six levels of abstraction recognized among computer programmers include: (1) the main purpose; (2) system architecture; (3) abstract data types; (4) algorithms and data structures; (5) source code; and (6) object code. *See id.* at 1343, n.10.⁵

After segmenting the program into successive levels, the Court filters out from each level of abstraction all elements that are not subject to copyright protection. *See Eng’g Dynamics*, 26 F.3d at 1344 (citing *Gates Rubber*, 9 F.3d at 837–38); *see also* 17 U.S.C. § 102(b). This filtration must occur *before* proceeding to the comparison step to determine substantial similarity and copyright infringement. *See Gates Rubber*, 9 F.3d at 843; *Eng’g Dynamics*, 26 F.3d at 1343 (“Third, the court should *then compare* the *remaining protectable* elements with the allegedly infringing program to determine whether the defendants have misappropriated substantial elements of the plaintiff’s program.”) (emphasis added).

After the filtration step, the remaining protectable elements (if any) are compared with

⁵ These levels of abstraction have been approved by at least the Tenth Circuit in *Gates Rubber*, whose refinement of the abstraction-filtration-comparison was adopted by the Fifth Circuit. *See id.* at 1343, n.10; *Gates Rubber*, 9 F.3d at 835; *Gen. Universal Sys.*, 379 F.3d at 142.

the accused program to determine if the accused program is substantially similar to the asserted works. *See Eng’g Dynamics*, 26 F.3d at 1342-43, 1347. That step is not at issue here.

2. The Burden to Complete Valid Abstraction and Filtration Is SASI’s.

For claims of infringement as to non-literal elements of a computer program, a plaintiff in this Circuit must “complete the *Altai* analysis necessary to evaluate claims that a program’s nonliteral elements were copied.” *Gen. Universal Sys.* at 144. “Without this analysis, there [will be] no evidence in the record to support [the plaintiff’s] claims that nonliteral elements were copied.” *Id.*; *see also* Ex. E (SCOTT ON INFO. TECH. L. § 2.51 (3d Ed. 2020-2 Supplement)) (“The burden is on the plaintiff to provide a complete *Altai* analysis.”).

SASI and its supporting amicus, Oracle Corporation,⁶ contend that WPL bears the burden to prove that any allegedly copied material is unprotectable. *See Oracle Br.* at 6-9; *Opening Br.* at 21-23. Relying on *Compulife Software Inc. v. Newman*, 959 F.3d 1288, 1301, 1305 (11th Cir. 2020), SASI and Oracle contend that, as a matter of policy, software copyright owners should not be required to identify the core protectable expression in their own copyrights. *See Opening Br.* at 21-22; *Oracle Br.* at 8. The Court should reject this position.

First, SASI’s policy preferences do not outweigh controlling Fifth Circuit authority: it is SASI’s burden to conduct a valid *Altai* analysis. *See Gen. Universal Sys.*, 379 F.3d at 144. This is because “[t]o establish copyright infringement, a plaintiff must prove ownership of a valid copyright and copying of constituent **elements of the work that are copyrightable**.” *Eng’g Dynamics*, 26 F.3d at 1340 (emphasis added).⁷ Moreover, the Fifth Circuit has expressly cast

⁶ SASI submitted an amicus brief supporting Oracle in Oracle’s litigation with Google.

⁷ *See also R. Ready Productions, Inc. v. Cantrell*, 85 F. Supp. 2d 672, 682 (S.D. Tex. 2000) (“In order to establish copyright infringement, the plaintiff must establish that defendant copied the protectable elements of plaintiff’s copyright.”).

~~XXXXXXDERRSEAKPURSKXXXXTROPROTECTXXXXORDER~~

doubt on the burden allocation SASI urges. *See Gen. Universal Sys.*, 379 F.3d at 143, n.26. The Fifth Circuit’s burden allocation keeps monopolies within their lawful scope by only comparing protectable elements. *See Eng’g Dynamics*, 26 F.3d at 1343 (“Filtration should eliminate from comparison the unprotectable elements”) (quotation omitted).

Second, SASI’s position pertains to “literal” copyright cases. This is not such a case. As SASI explains, “[t]his is not a case of copyright infringement based upon direct copying of source code Instead, the allegations in this case are that WPL stole SAS’s input formats and output designs—**non-literal elements** of its software.” Dkt. 266 at 6 (emphasis added). In contrast, *Compulife* was a literal copyright case. There, the source code from the asserted work (registered HTML source code) was copied by the defendant (by “scraping” the plaintiff’s database). *See id.* at 1299-1304. The *Compulife* court expressly did not apply the *Altai* test and confined its opinion to “literal” copyright—i.e., registered code. Only in that circumstance does the decision place the burden on a defendant to demonstrate that copied elements are unprotectable. *See id.* at 1304-05 (citing *MiTek Holdings, Inc. v. Arce Eng’g Co.*, 89 F.3d 1548, 1555 n.16 (11th Cir. 1996), which explained that the *Altai* test is for claims over “a nonliteral element, not copying of computer code (a literal element)”).⁸ SASI asserts non-literal elements, and to meet its burden, it was required to define its protectable expression through proper filtration.⁹ SASI has not even attempted to carry that burden.

⁸ From a policy perspective, applying the *Compulife* burden allocation (on a defendant to show unprotectability) to literal infringement and the *General Universal* burden allocation (on a plaintiff to show protectable elements copied) makes intuitive sense. Copying of source code is likely to result in copying of expression, while copying nebulous non-literal elements tends toward the highly functional. But in any case, *Compulife*’s literal infringement burden allocation does not apply as literal infringement is not at issue in this case.

⁹ Unlike some other circuits, the Fifth Circuit treats filtration as a copyrightability issue—part of a plaintiff’s burden—rather than as a defense to infringement. *See Lexmark Intern., Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 557-59 (6th Cir. 2004) (recognizing circuit split and Fifth

3. Any Registration Presumption Is Insufficient to Carry SASI's Burden.

While copyright registrations do provide a presumption of validity in the copyright of a work *as a whole*, this presumption is irrelevant here.¹⁰ “The mere fact that a work is copyrighted does not mean that every element of the work may be protected.” *See Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 348 (1991); *see also Kepner-Tregoe*, 12 F.3d 527, 533 (5th Cir. 1994) (noting that the “mere fact” that plaintiff’s works were “copyrighted does not mean that all aspects of those materials are automatically protected.”) (footnote omitted).¹¹

The lack of any presumption as to any specific element is relevant here. A registration reflects only that “the material deposited constitutes copyrightable subject matter.” 17 U.S.C. § 410(a). It is “*prima facie* evidence of the validity of the copyright and *of the facts stated in the certificate.*” *Id.* at 410(c) (emphasis added). [REDACTED]

[REDACTED]

[REDACTED] Thus, the Copyright Office made, at most, a determination of the source code’s copyrightability. *See* 17 U.S.C. § 410(a); *see also Kepner-Tregoe*, 12 F.3d at 533. Nothing in the registration certificates, however, mentions the asserted “collection” of the “input formats” or “output designs,” or how they are supposedly sequenced, structured, and organized. Because the

Circuit approach); *Mason v. Montgomery Data, Inc.*, 967 F.2d 135, 138, n.5 (5th Cir. 1992); *see also Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1358 (Fed. Cir. 2014) (recognizing circuit split). The Fifth Circuit’s decision to treat filtration as an issue of copyrightability supports its decision to allocate the filtration burden to non-literal copyright plaintiffs.

¹⁰ The presumption is also only a rebuttable presumption. *Lakedreams v. Taylor*, 932 F.2d 1103, 1108, n.10 (5th Cir. 1991) (“Regardless, the certificate does not create an irrebuttable presumption of originality.”); *Norma Ribbon & Trimming, Inc. v. Little*, 51 F.3d 45, 47 (5th Cir. 1995).

¹¹ *See also Oracle Am., Inc. v. Google Inc.*, 810 F. Supp. 2d 1002, 1009 (N.D. Cal. 2011) (“[A] certificate of registration may entitle its holder to a presumption of copyright validity as to the registered work” but not “a presumption of originality as to specific elements of a registered work”) (emphasis in original).

non-literal elements are not present in the material deposited with the applications, and the non-literal elements are not otherwise described, identified, or stated in the certificates, the registrations do not serve as *prima facie* evidence of the copyrightability of the non-literal elements.¹² *See* 17 U.S.C. § 410(c); *see also* *Kepner-Tregoe*, 12 F.3d at 533.

4. The Court May Resolve All Disputed Facts in Connection with Its Abstraction and Filtration Analysis.

The Court has correctly recognized that copyrightability, including the abstraction and filtration analyses, include questions of law. *See* Dkt. 436 at 1-2; *accord Harbor Software, Inc. v. Applied Sys., Inc.*, 925 F. Supp. 1042, 1046 (S.D.N.Y. 1996). But the Court’s role involves more than simple resolution of legal questions. District courts within the Fifth Circuit have instead determined that issues of copyrightability—for questions of both law and fact—should be resolved by the judge and not the jury. *See, e.g., Compaq Computer Corp. v. Ergonome, Inc.*, 137 F. Supp. 2d 768, 775, n.3 (S.D. Tex. 2001) (concluding that questions of copyrightability, both fact and law, should be determined by the court, not the jury); *Jane Envy, LLC v. Infinite Classic Inc.*, SA:14-CV-065-DAE, 2016 WL 797612, at *4 (W.D. Tex. Feb. 26, 2016) (concluding that copyrightability should be evaluated at the summary judgment stage); *see also* Ex. G, C.J. Sprigman & S.F. Hedrick, *The Filtration Problem in Copyright’s “Substantial Similarity” Infringement Test*, 23 LEWIS & CLARK L. REV. 571, 587 (2019) (“But there is a deeper problem - many laypeople do not share the instrumentalist instincts that underlie the copyright ‘balance,’ but instead tend to react to copying as categorically wrongful.”).

¹² Even if the presumption extended to copyrightability of elements, the presumption would at most shift the burden of *production* to WPL to present evidence rebutting the presumption. See FED. R. EVID. 301. The burden of persuasion would never shift since the element of copyrightability is an element of SASI's claim. *Id.*; cf. *Eng'g Dynamics*, 26 F.3d at 1340. WPL has presented more than sufficient evidence to shift any burden back to SASI.

B. On This Record, There Is No Core Protectable Expression.

SASI's affirmative claim of copyright infringement is hopelessly ill-defined. Although SASI contends that WPL has somehow coopted the "structure, sequence, and organization," of the Asserted Works, SASI has *never* articulated how the "input formats" or "output designs" are supposedly sequenced, structured, and organized in its own Asserted Works. SASI's dogged refusal to complete a valid AFC analysis compounds these issues.

Although SASI was required to identify abstraction layers divided "into successive levels of generality," Dr. Storer instead identified five disjointed aspects of the Asserted Works labeled them abstractions: (1) main purpose of the program; (2) interface mechanism; (3) input formats; (4) output designs; and (5) naming and syntax.¹³ Other than the program's main purpose, these "layers" appear to be cherry-picked characterizations of elements of the SAS Language and, to a much lesser degree, the SAS software. Dr. Storer's testimony confirms that his analysis does not divide the program as a whole into successive levels of generality.¹⁴ This is not surprising, as Dr. Storer was not even asked to review the Asserted Works in this case. *Id.* at 396:10-21.

On filtration, Dr. Storer simply looks the other way concerning a host of unprotectable elements. SASI seeks to wield its copyrighted code to monopolize all of the following unprotectable elements:

(1) public domain elements such as SAS 76; (2) elements dictated or contributed by user's SAS language programs; (3) facts and data; (4) elements taken from third parties like the PL/I and SPSS languages and programs; (5) mathematical and statistical algorithms, calculations, variables, and measurements; (6) well-known ideas or templates for graphically displaying data, like tables, graphs, charts, plots, colors, and fonts; (7) expressions that have merged with idea, facts, or functionality;

¹³ See Storer Copyright Rpt. at 26-27.

¹⁴ See, e.g., Ex. H (Storer Dep. Tr.) at 277:25-278:4 (interface mechanism "layer" not a "collection" of inputs/outputs at a higher level); *id.* at 278:5-9 (interface mechanism not a higher level of abstraction than the "collection" of inputs/outputs); *id.* at 278:10-15 (interface mechanism is not a more specific level of abstraction of the main purpose of the SAS software).

~~XXXXXXXXXXXXXX ORDER X P X R S U A N T X O X P R O T E C T X X E X O R D E R~~

(8) scenes à faire elements in statistical and mathematical analysis as well as computer science and computer programming; and (9) short phrases, fragmentary words, and abbreviations.

See Jones Decl. Ex. 1 (Jones Rpt.) at § 8.2.2 (¶¶ 108-214). Dr. Storer concedes that the critical portions of his analysis were not subjected to any filtration.¹⁵

1. The Court Can Find No Core Protectable Expression Because SASI Failed to Meet Its *Altai* Burden.

Despite it being SASI's burden, SASI cannot even explain to the Court the nature of its claimed copyright or identify the protectible expression that WPL has allegedly copied. Nor has SASI made any endeavor to ensure that its copyright assertion does not overreach and invite comparison with unprotectible elements. Under controlling law, the Court may find on this record that there is no core protectible expression because SASI has failed to offer admissible evidence of any. *See Supra Section V.A.2.*¹⁶

This result is warranted. SASI had numerous opportunities to develop its case. WPL served interrogatories requesting SASI's position on filtration, wrote letters to SASI, filed a motion to compel (Dkt. 129), and filed a renewed motion to compel (Dkt. 219). SASI shirked its discovery obligations, leaving WPL thinking that it would first learn of SASI's filtration positions at the expert report stage. But SASI did not even do this. Dr. Storer's report had no

¹⁵ See Dkt. 441-2 (Storer Decl.) at 7 n.1 ("I did not filter out elements from [the input formats]"); *id.* at 12 ("[T]he output designs were not subject to filtration . . ."); *id.* at 12 ("[N]aming and syntax of individual PROCs, statements, options, formats [sic] informs, global statements, access engines and other elements . . . should not be subject to filtration.").

¹⁶ And here, the "application of the abstraction/filtration/comparison test in the instant case requires expert knowledge in the area of computer programming—a level of knowledge that the Court believes requires a fundamental understanding of computer programming." *See Pepper v. Int'l Gaming Sys., LLC*, 312 F. Supp. 2d 853, 861 (N.D. Miss. 2004). Without a proper AFC analysis from Dr. Storer, SASI will have "no expert to help explain to the court and the jury how to apply the abstraction/filtration/comparison test to the computer programs in the instant case—or anything other technical aspect of the plaintiff's case." *See id.* (granting summary judgment after having stricken expert for failure to comply with Rule 26(a)(2)).

valid AFC analysis—he substituted filtration with *Feist* creativity evidence. *See* Ex. H (Storer Dep.) at 156:22-157:25; 58:16-160:14; 160:19-164:18. WPL then served a rebuttal expert report with a robust filtration analysis, filed motions for summary judgment and to exclude Dr. Storer that pointed out the lack of filtration, and responded to SASI’s motions pointing out that SASI had not met its burden to do a proper filtration. SASI still did not filter.

The Court then ordered SASI to submit filtration evidence. Dkt. 436. Instead of complying, SASI argues against it—claiming that it was not required to do so. According to SASI, it can sidestep filtration and the Court’s order merely by asserting that its software includes “creative choices,” *see* Dkt. 441 at 6-11, 21-23, 29-30—as if “creative choices” is an incantation to charm doctrines of public domain, merger, *scènes à faire*, etc. into nonexistence.

SASI’s resort to generic “creativity and expressiveness” arguments is insufficient for a case about non-literal program elements. *See Computer Mgmt.*, 220 F.3d at 400 (in case involving non-literal elements, “[w]e use the ‘abstraction-filtration’ method to determine copyright protection”). In such cases, the plaintiff must complete a valid AFC analysis. Indeed, a district court in the Fifth Circuit recently rejected a plaintiff’s attempt to only show evidence of originality or creativity. *See M-I LLC*, 2020 WL 4549210, *3. Particularly given SASI’s foregone opportunities, the Court can and should do the same here.

Furthermore, SASI has not articulated where the “input formats” or “output designs” are within the Asserted Works, much less how they are sequenced, structured, and organized there. Dr. Storer’s report includes a diagram that purports to illustrate how *World Programming’s* WPS code base is organized and structured, but not the SAS System. *See* Storer Rpt. at Ex. 5. Similarly, Dr. Storer’s report includes an exhibit excerpting WPS file paths and comparing them to excerpts of a SAS manual. *See* Storer Rpt. at Ex. 4. However, the SAS manual is not an

Asserted Work. And Dr. Storer’s report does not establish any link between any manual and the “input formats” or “output designs” or their sequence, structure, and organization in the software.

Accordingly, SASI’s refusal to complete a valid *Altai* analysis, or even describe the structure and organization of the elements it purports to assert, leaves WPL and the Court facing an amorphous set of allegations that cannot be prepared for a substantial similarity comparison.

2. There Is No Core Protectable Expression Related to the Asserted Works’ “Collection of Input Formats” or “Naming” of SAS Language syntax.

The first “non-literal element” that SASI identifies as allegedly subject to protection is a “collection of input formats.” Like the rest of its case, this assertion remains fuzzy. In some places, SASI claims that *each* input format is “creative expression,” and “each of SAS’s many PROCs” are like Oracle’s copyrighted API packages. *See, e.g.* Dkt. 441. But in other places, SASI concedes that none of its “input formats” are individually protectable. Dkt. 441 at 27. Instead, SASI claims protection in “the selection and arrangement in the SAS input formats *as a whole.*” *Id.* (emphasis added). But while SASI uses words like “selection” and “arrangement,” it never indicates whether the “collection” of input formats is a compilation of uncopyrightable material, a creative structure or organization, or both. Under any formulation, no core protectable expression exists in the alleged “collection of input formats.”¹⁷

First, what SASI characterizes as a “collection of input formats” is actually the SAS computer programming language.¹⁸ The SAS Language consists of a syntax for writing computer programs. *See* Jones Decl., Ex. 1 (“Jones Rpt.”) at § 5.2. It was developed in the

¹⁷ SASI’s “individual” naming element appears to be the individual words of the SAS Language.

¹⁸ While SASI maintains the façade of not alleging infringement of a language, this is belied by its own witnesses and filings. *See, e.g.*, Ex. A (Goodnight Dep. Tr.) at 51:3-15 (admitting position that SASI “has a copyright to the SAS language itself”); *id.* at 77:25-78:6 (“It is my belief that we do own the language, and it is copyrighted.”); *see also* Dkt. No. 264 at 15 (“Thus, WPL itself claims that WPS ‘[p]arses SAS Language input files,’””) (citation omitted and emphasis added).

1960's through government grants and is not itself copyrighted by SASI or any of the "hundreds" of individuals around the country who developed the language. *See* Jones Rpt. at Ex. I (SAS 76) at Acknowledgements. SASI has conceded the SAS Language is free to use by the public without a license.¹⁹ Indeed, while SASI draws heavily from the *Oracle v. Google* case, one thing was not subject to debate there: Java, a programming language created by Oracle's predecessor, was "open and free for anyone to use." 750 F.3d at 1353. The SAS Language is similarly unprotectable here. *Accord* Dkt. 441, Attachment 1, *Cisco v. Arista Networks, Inc.*, Slip Op. at 11 ("... [t]he Court will instruct the jury that there is no protection for syntax.").²⁰

Second, SAS Language elements are not expressed in the Asserted Works in the way SASI would have the Court believe (if these elements are even expressed in the Asserted Work to begin with). For example, SASI argues that PROC MIXED "has the following structure as input format options as shown in the SAS manual" and then includes the image on the left:²¹

What SASI shows in its Brief:

SAS/STAT 14.1 User's Guide: The MIXED Procedure
 Product: SAS/STAT
 Procedure: Proc Mixed
 PROC MIXED < options >;
 BY variables ;
 CLASS variable <(REF= option> ... <variable <(REF= option> >> </global-option
 CODE < options >;
 ID variables ;
 MODEL dependent < fixed-effects > </ options >;
 RANDOM random-effects </ options >;
 REPEATED < repeated-effect > </ options >;
 PARMs (value-list) ... </ options >;
 PRIOR < distribution > </ options >;
 CONTRAST 'label' < fixed-effect values ... >
 <| random-effect values ... > ... </ options >;
 ESTIMATE 'label' < fixed-effect values ... >
 <| random-effect values ... > </ options >;
 LSMEANS fixed-effects </ options >;
 LSMESTIMATE model-effect lsmeestimate-specification </ options >;
 SLICE model-effect </ options >;
 STORE < OUT=>item-store-name </ LABEL='label' >;
 WEIGHT variable ;

What is actually in the SAS Manual:

Syntax: MIXED Procedure
 The following statements are available in the MIXED procedure:
 PROC MIXED < options >;
 BY variables ;
 CLASS variable <(REF= option> ... <variable <(REF= option> >> </global-options >;
 CODE < options >;
 ID variables ;
 MODEL dependent = < fixed-effects > </ options >;
 RANDOM random-effects </ options >;
 REPEATED < repeated-effect > </ options >;
 PARMs (value-list) ... </ options >;
 PRIOR < distribution > </ options >;
 CONTRAST 'label' < fixed-effect values ... >
 <| random-effect values ... > ... </ options >;
 ESTIMATE 'label' < fixed-effect values ... >
 <| random-effect values ... > </ options >;
 LSMEANS fixed-effects </ options >;
 LSMESTIMATE model-effect lsmeestimate-specification </ options >;
 SLICE model-effect </ options >;
 STORE < OUT=>item-store-name </ LABEL='label' >;
 WEIGHT variable ;

¹⁹ See, e.g., Ex. I (Langston Dep. Tr. (30(b)(6), Jan. 29, 2014) at 26:3-9, 32:8-18; Ex. F (Rubendall Dep. Tr. (30(b)(6), Feb. 26, 2020) at 266:15-267:1, 262:18-267:1.

²⁰ See also *SAS Inst. Inc. v. World Prog'g Ltd.*, 64 F. Supp. 3d 755, 775–76 (E.D.N.C. 2014).

²¹ Compare Dkt. 441 at 12 with Ex. J at 6058 (PROC MIXED, explaining summary descriptions of functionality and syntax for statements is provided in alphabetical order); cf. *Feist*, 499 U.S. at 363 ("But there is nothing remotely creative about arranging names alphabetically . . .").

SASI's purported "structure as input format options"—or as the omitted portion of its manual explains: the "syntax"—does not appear in the actual SAS software in this suggested form. *See* Jones Decl. at ¶¶ 5-6, 9-11. In other words, SASI has not shown that when the SAS software is abstracted from the literal code all the way up to the main idea, there exists a layer of abstraction that would include this "structure," or even a list that includes all of these elements, as presented here. *See* Jones Rpt. at Ex. E; Jones Decl. at ¶¶ 9-11.²² [REDACTED]

[REDACTED] *See* Jones Decl. at ¶¶ 9-11.

But tellingly, [REDACTED]

[REDACTED]. *See id.*

SASI's failure to identify such elements is not merely semantics—it goes to the heart of what the alleged expression might be. [REDACTED]

E; Jones Decl. at ¶¶ 9-11. In these instances, [REDACTED]

[REDACTED] *See id.* SASI's

assertions to date, however, do not recognize or acknowledge the organization of the code modules and data structures in the SAS software. *See* Jones Decl. at ¶¶ 9-11. SASI instead appears to contend that each PROC is a self-contained unit, [REDACTED]

[REDACTED]. *See* Jones Decl. at ¶¶ 9-11.

²² This further highlights the flaws in Dr. Storer and SASI's analysis—by failing to perform a proper abstraction that begins with the code and goes to the main idea, they lost sight of what the actual copyrighted work was. *See* Jones Rpt. at § 8.2.1, ¶¶ 91-106. A consequence of this failure is that SASI now asserts nebulous, uncopyrightable ideas like "structures as input format options" that do not even exist in the Asserted Works.

By contrast, the “input formats” in *Engineering Dynamics* were not aspects of a computer programming language. Instead, they were graphical representations of “keypunch cards”:

See *Eng’g Dynamics*, 26 F.3d at 1352. There is a preset, fixed, and consistent appearance of the card, and there is narrative and explanatory text to help “instruct the user to place specific kinds of information in a specific place on the card.” *Eng’g Dynamics*, 26 F.3d at 1338. As the Fifth Circuit explained, the “input and output formats for SACS IV are quasi-textual; while they guide the user in performing a series of sophisticated structural analyses, they consist of a series of words and a framework of instructions that act as prompts for the insertion of relevant data.”²³

See id. at 1342. Unlike, *Engineering Dynamics*, the alleged “input formats” here are not a user interface, and there is no user interface with a similar preset, fixed, and consistent appearance requiring placement of information in precise locations. There are no instructions or prompts

²³ In any event, SASI's strenuous attempt to shoehorn into *Engineering Dynamics* is heedless. Following the panel decision, the Fifth Circuit issued a supplemental opinion to confirm that “[t]he panel did not say that in any case involving user interface the fact that the ‘author’ has selected from among possible formats is dispositive.” *See* 46 F.3d 408 (5th Cir. 1995). It then remanded the case with instructions to determine whether any protectable expression remained after filtration of the inputs and outputs. *Eng’g Dynamics*, 26 F.3d at 1351. On remand, the court performed the required filtration—the step SASI contends should be disregarded—and found nothing left. *See* Ex. K (SJ Opinion). Judgement was entered for the defendant on this copyright claim. *See* Ex. L.

that direct a user to enter information into a specific place. Instead, the primary interface of the Asserted Work is a generic code editor that provides a blank space to write free-form code.²⁴

Third, even if the Court were to adopt SASI’s formulation of the free-to-use programming language as a set of protected “input formats,” filtration is still required. SASI’s brief enumerates components of its asserted “input formats,” *see* Opening Br. at 11 (“PROCs, statements, options,” etc.), yet even a single public domain filter demonstrates the lack of core protectable expression. *See* Appendix A. The same is true on a more granular level with respect to the sole example provided by SASI, PROC MIXED. *See* Appendices B & F.

Finally, SASI’s undeveloped “compilation” also fails to evidence any core protectable expression. Copyrights in compilations—like an alleged “collection” of input formats—provide only “thin” copyright protection; only the selection, coordination, and arrangements of the materials could be protectible. *See Eng’g Dynamics*, 26 F.3d at 1346 (quoting *Feist*, 499 U.S. at 359, 349). The specific selection, coordination, and arrangement must still meet *Feist*’s “modicum of creativity” standard, and it must not be otherwise barred by doctrines of merger, *scenes a faire*, public domain, and the like.²⁵

At bottom, SASI is trying to misuse its copyrights to monopolize functionality. It does not want anyone else to be able to parse or read formats consisting of SAS Language elements.

²⁴ *See* Jones Rpt. at ¶ 286, Ex. F at 1, 4 (showing SAS editor).

²⁵ Here, there is no creativity or originality in the alleged “collection of input formats,” which is simply a compilation of all SAS Language elements. *See Feist*, at 363 (no creativity by compiling all the people who had applied for telephone service). Moreover, any software wanting to “make use” of the SAS Language would necessarily include significant aspects of the alleged compilation. In that way, any expression in the selection would merge with the idea of the language itself. In *Cisco v. Arista*, the court made clear that the compilation was “comprised largely of unprotectable elements and thus subject to thin protection.” *Cisco Sys., Inc. v. Arista Networks, Inc.*, 5:14-cv-05344-BLF, Doc. 633 at 18 (N.D. Cal. Nov. 10, 2016). The court did not address whether any expression in a compilation merged with the underlying idea because merger is an infringement defense in the Ninth Circuit. *Id.* at 6.

But this is not copyrightable expression. In fact, the Fifth Circuit explained in a supplemental opinion that *Engineering Dynamics* “cannot properly be read to extend . . . to the practice employed by users of programs of analyzing application programs to ‘read’ the file formats of other programs.” 46 F.3d 408, 410 (5th Cir. 1995). Accordingly, SASI has not established that the SAS Language elements are protectable under copyright law. There is no relevant core protectable expression in its “input formats.”

3. There Is No Core Protectable Expression Related to the Asserted Work’s Relevant “Output Designs.”

Copyright in a computer program can extend to the program’s menus and screen displays. *See Gen. Universal Sys.*, 379 F.3d at 142. This is so because the program code contains fixed expression that produces the screen displays. *See* U.S. Copyright Office, Compendium of U.S. Copyright Office Practices § 721.10(A) (3d ed.) (2017).²⁶ In the typical scenario, these menus and screen displays are those generated when the code of the copyrighted program runs. *Id.* The SAS System has such menus and screens generated by the copyrighted code—but that is not what SASI is asserting. Rather, the “output designs” at issue are outputs generated by third parties (i.e., users) who design their own computer programs and run them using SAS’s software. SASI has no copyright in the outputs created by others— [REDACTED]
Ex. F (Rubendall Depo Tr.) at 262:18-263:5. The claimed “collection of output designs” is not a collection of anything within the Asserted Works because the outputs are not fixed there, and do not exist, until a third party submits a program specifying what the output should entail. *Id.* at 185:15-21; Ex. Jones Rpt. at § 8.2.3; *see also* 17 U.S.C. § 102(a). The reason SASI cannot define its “collection” of outputs is because it escapes definition—an infinite number of outputs

²⁶ Available at <https://www.copyright.gov/comp3/chap700/ch700-literary-works.pdf>.

exist based on the infinite number of programs that could be designed and run by users.

At the periphery, copyright protection can extend to output that is partially dictated by user action—but only where a user’s role is marginal in bringing about the generation of output from a copyrighted programs such that the output reflects the program’s contents.²⁷ That is not the case here. The “output designs” do not reflect the program’s contents and thus fail the fixation requirement. *See* 17 U.S.C. § 102(a). And SASI has presented no evidence establishing that the contribution in generating outputs is only marginal. Rather, the evidence shows the opposite. *See* Jones Rpt. at § 8.2.3. Mr. Rubendall, SASI’s 30(b)(6) witness on copyright issues,

[REDACTED] *See* Ex. F at 210:14-211:13. He further conceded that, [REDACTED]

[REDACTED] *Id.* at 221:1-20

SASI’s other witnesses admitted the same.²⁸ Indeed, Dr. Storer conceded that the allegedly infringed outputs are “governed by the program that the user – the way *the user chose to express themselves* in the form of that program resulted in a certain output.” Ex. H at 282:16-22. This evidence overwhelmingly confirms that the output owes its origin to the user, not the SAS software, and is not fixed in a tangible medium of expression unless a user creates and executes its own unique program. *See Eng’g Dynamics*, 26 F.3d at 1346; *Feist*, 499 U.S. at 359, 349; *see*

²⁷ Compare *Digital Drilling Data Sys. LLC v. Petrolink Servs. Inc.*, No. 4:15-CV-02172, 2018 WL 2267139, at *6 (S.D. Tex. May 16, 2018) (user’s role marginal when merely filled in pre-made forms); *Torah Soft Ltd. v. Drosnin*, 136 F. Supp. 2d 276, 283 (S.D.N.Y. 2001) (user’s role marginal when merely entered search terms), with *Rearden LLC v. Walt Disney Co.*, 293 F. Supp. 3d 963, 970–71 (N.D. Cal. 2018) (users’ role more than marginal even when program did a “significant amount of work” to transform two-dimensional information into three-dimensional outputs); *Design Data Corp. v. Unigate Enter., Inc.*, 847 F.3d 1169, 1173 (9th Cir. 2017) (finding plaintiff failed to set forth evidence that users’ input was marginal when data was used by program that also used building and engineering codes to create steel detailing files).

²⁸ *See* Ex. N (Whitcher) at 223:5-224:23; Ex. A (Goodnight) at 60:8-10, 60:23-61:4.

~~XXXXXXXXXXXXXXAXXXXXXXXXXXXXX~~

also Appendix C (comparing SAS Language user program to an “output design” asserted by SASI as its copyrighted expression).

SASI cannot create copyrightability where none exists based on its argument about “default” designs in the outputs. Defaults or not, the users’ input here is unlike the entry of simple search terms like in *Torah*, or the filling in of pre-made forms like in *Digital Drilling*. Instead, this case is more akin to *Rearden LLC v. Walt Disney Co.*, 293 F. Supp. 3d 963, 970–71 (N.D. Cal. 2018), where the software enables a wide range of possibilities, all up to the user, of what output can be generated. The “defaults” do not negate the fact that it is users’ contributions that dominate the output.²⁹ The SAS user interface is a typical code or text editor with blank space. *See* Jones Rpt. at ¶ 286, Ex. F at 1, 4. The users can write code that addresses a wide range of possibilities in terms of analyzing data and solving mathematical and statistical problems. That includes choosing or modifying the “default” at various levels. *See id.* at ¶¶ 224, 228; Jones Decl. at ¶ 13 (explaining these are not true “defaults”). Both are contemplated in the Asserted Works, confirming that a user’s contribution is much more than marginal. Also, the output of a SAS Language program is heavily dictated by the data the user feeds into the SAS system software. *See* Jones Rpt. at ¶¶ 225–27 (explaining user’s selection of data, variables, how to encode results, which mathematical procedures to run and in what order). The numerical analysis that the SAS software performs on the dataset might be a “significant amount of work”—but that “cannot be enough.” *See Rearden*, 293 F. Supp. 3d at 970–71. The users’ “role

²⁹ *See also* Ex. F at 210:14–211:13 (after highlighting “all of the outputs” to indicate they were dictated by the SAS language program, testifying that in that example, the output is from the user’s SAS language program, the SAS language program is a significant component to the output, and the user of the SAS language program “explicitly stated what they wanted to happen”); *id.* at 241:21–242:10; *see also* Jones Rpt. at ¶ 228 (explaining users of SAS software are sophisticated and choose to accept default settings and that this is not a take-it-or-leave scenario).

in creating the end-product is not so ‘marginal’ that the output reflects the program’s contents.”

See id. (internal quotations omitted).

Trying to prove otherwise, SASI has identified a simplistic SAS program invoking PROC MIXED. *See* Dkt. 441 at 13. It is worth noting that SASI has omitted over half of the user’s program (ostensibly to portray it as marginal). In any event, even in that example, the user must select which PROC to invoke, must define and provide input data on which the statistical analyses are to be conducted, must select and define the classes, must select and define the model, must select and define the effects (“random”), and must decide to include a title and invoke TITLE. *See* Jones Decl. at ¶¶ 12-13. Also, a “mixed” model analysis is a conventional type of statistical analysis that mixes statistical models, with known analyses, variables, and outputs.³⁰ SASI is left with a selection of a conventional description for a statistical analysis, and conventional constituent data points incident to that analysis specified by the user in designing its program. That does not render the user’s contribution merely marginal.

Finally, as with the “collection of input formats,” SASI has not identified or described the expression of the “collection of output designs” in the form in which it appears in the Asserted Works. As before, when the SAS software is abstracted from code to main idea, there is no layer of abstraction that would include this “output design” in the form SASI suggests—and as noted, it does not exist until a *user* program, not the SAS software, is created and executed. *See* Jones Rpt. at § 8.2.3, Ex. E; Jones Decl. at ¶¶ 12-13. These “output designs” do not exist solely and neatly on a per-PROC basis (as SASI pretends), and SASI has not identified any modules or data structures in the software to demonstrate otherwise. *See* Jones Decl. at ¶¶ 12-13 .

³⁰ *See* Ex. O, Wolfinger, R. D., Tobias, R. D., and Sall, J. (1991). “Mixed Models: A Future Direction.” In *Proceedings of the Sixteenth Annual SAS Users Group Conference*, 1380–1388. Cary, NC: SAS Institute Inc., *cited in* Jones Rpt. Ex. C at 35.

In sum, SASI has not shown that any copyright over its SAS system software extends to the outputs generated as a result of the execution of users' SAS Language program. Such outputs are neither original to SASI nor sufficiently fixed to be copyrightable. The elements described in this section should all be filtered out in the filtration step described above. This, combined with all of the other required filtration, including public domain (*see* Appendix D), demonstrates that SASI has identified no core protectable expression in "output designs."

4. *Google v. Oracle* Does Not Save SASI From Its Failure to Prove Any Protectable Expression.

SASI (and Oracle) expend much energy trying to align this case to *Oracle v. Google*. The analogy is tortured. That case involved two discrete allegations: (1) copying literal elements of software (i.e., "declaring source code"); and (2) copying non-literal elements of software (i.e., "structure, sequence, and organization of API packages"). *See Oracle*, 750 F.3d at 1356. SASI is not alleging copying of literal elements of software here, thus *Oracle*'s analysis as to "declaring source code" is irrelevant to SASI's allegations of copying non-literal elements.

With respect to analysis of non-literal elements, *Oracle* did not find that a programming language, its "organizational structure," or its grammar were copyrightable. *See* 750 F.3d at 1367-68. Rather, the Federal Circuit reasoned that having copied the literal declaring source code at issue, Google had not used just the structure and grammar of the APIs, but the specific way it was implemented at Oracle. *See id.* More generally, Google's admission of copying source code factored heavily into the analysis concerning non-literal elements. *See, e.g.*, 750 F.3d at 1356 (explaining that when Google copied source code, it also copied sequence and organization of API packages), 1365 (distinguishing case on the basis that Google copied source code verbatim). Here, there is no allegation, much less concession, of source code copying. Finally, *Oracle* did not consider significant bars to copyrightability that feature prominently in this case, including

that SASI has presented no evidence of fixation of the asserted “non-literal” elements in the Asserted Works, the fact that the plaintiff’s expert has not reviewed the Asserted Work, that the “non-literal” elements are in public domain SAS 76, and that “non-literal” elements originated from government grants preceding SASI and involved “hundreds” of non-SASI authors.³¹

C. A Proper *Altai* Analysis Would Proceed Like Dr. Jones’s.

The abstraction step focuses on “how to describe the levels of generality ascending in computer programs from the literal code to the most general ‘idea’ of the program itself.” *Eng’g Dynamics*, 26 F.3d at 1343, n.10. “[A] computer program can often be parsed into at least six levels of generally declining abstraction: (i) the main purpose, (ii) the program structure or architecture, (iii) modules, (iv) algorithms and data structures, (v) source code, and (vi) object code.” *Gates Rubber*, 9 F.3d at 835; *see also* *Eng’g Dynamics*, 26 F.3d at 1343, n.10.

Rather than select disjointed aspects of the Asserted Works, Dr. Jones applies the *Gates Rubber* layers. As applied to the Asserted Works, those layers are (Jones Rpt. at § 8.4):

- (1) the main purpose of the program – the main purpose or ultimate function of the program is to provide ways of performing statistical analysis and view the results, including by letting users execute programs written in the SAS Language;
- (2) the program architecture – the program architecture is the overall set of components and relationships between them that work together to operate in a certain way;
- (3) modules – the data modules contain algorithms and data structures and represent functions or operations that can be carried out to accomplish a given task, such as an operation to read and store input data;
- (4) algorithms and data structures – the algorithms are the processes or sets of rules that are followed in an operation to solve a problem, such as a particular formula or set of steps to calculate a regression. The data structures are the

³¹ *Oracle* also treated almost all issues of copyrightability (e.g., merger and *scènes à faire*) as defenses to a claim of infringement, not bars to copyrightability. *See Oracle*, 750 F.3d at 1358. In contrast, the Fifth Circuit applies filters at the copyrightability inquiry. *See Lexmark*, 387 F.3d at 557-59; *Mason*, 967 F.2d at 138, n.5.

stores of values and attributes about them or relationships between them, such as an object with information about a dataset;

(5) source code – the source code is the set of human-readable code or instructions written by programmers or developers that, when compiled into object code, comprise the object code that is run and launches the SAS software; and

(6) object code – the object code is the machine-readable code or instructions (as would be in an executable) that when run launches the SAS software and allows the user to interact with it.

At the filtration step, the main purpose and the program architecture layers are filtered out. The main purpose of the program is unprotectable as purely an idea or purely functional. With respect to the program architecture, some of its elements, such as its structure, sequence and organization, could potentially be protectable. However, SASI has not presented any evidence of its computer program architecture. *See, e.g.*, Storer Rpt. at Ex. 5 (purporting to contain illustration of WPS program organization and structure but not providing similar illustration for SAS software). Elements of SAS's computer program source code and object code might also be protectable, but SASI is not alleging infringement of its source code or object code. This leaves the layers for modules and algorithms and data structures.

Of these two remaining layers, it is necessary to filter out unprotectable elements from “each particular level.” *Eng'g Dynamics, Inc.*, at 1344. The filters at each level include public domain elements, open source and third-party elements, elements that are not original to SAS (such as PL/I, SQL, or SPSS), mathematical or statistical analysis elements, factual and data elements, process / system / method elements, ideas like the SAS programming language itself, conventional graphical display elements (such as tables, graphs, plots, colors, or fonts), merged elements, scenes à faire elements, and short phrase elements. *See generally* Jones Rpt. at § 8.2. Numerous filters must be applied to SASI’s “non-literal” elements, including the following:

Filtration	
WPL	SAS
Public Domain works	<i>No evidence of filtration.</i>
Names, Abbreviations, Shorthand	<i>No evidence of filtration.</i>
“Facts” Such as Mathematics	<i>No evidence of filtration.</i>
Portions Unoriginal to SASI & User input/user data	<i>No evidence of filtration.</i>
<i>Scènes à faire</i>	<i>No evidence of filtration.</i>
Ideas (Merger)	<i>No evidence of filtration.</i>
Well-known ideas or templates for graphically displaying data, like tables, graphs, charts, plots, colors, and fonts	<i>No evidence of filtration.</i>

Every aspect of SAS 76 must be filtered out.³² Here, even an alleged “selection” would not pass the *Feist* creativity standard as it is conventional and routine to take prior elements forward in subsequent software versions.

Almost all of the types of terms that Dr. Storer identifies are names, abbreviations, or shorthand for statistical and mathematical concepts, procedures, calculations, variables, and measurements that are conventional or dictated by standards in the fields of statistical analysis or in computer programming.³³ *See Taylor*, 54 F. App’x 794 (“Copyright Act does not protect fragmentary words or short phrases”); *Eng’g Dynamics*, 26 F.3d at 1345 (“Likewise, abbreviations for terms, dictated by necessity or industry standard, are uncopyrightable by themselves.”). These terms are also often simply reflective of or descriptive of the functionality assigned to them. For example, PROC MEANS calculates mean values. PROC PRINT prints a dataset. Similarly, other terms like “header,” “method,” “info”, “data,” “order,” and “label” are short phrases or shorthand that reflect their functionality, industry standards, or programming standards and practices.³⁴ *See Computer Mgmt.*, 220 F.3d at 401. This must all be filtered. *See, e.g.*, example filtration at Appendix B.

³² *See, e.g.*, example filtration at Appendix A through E.

³³ *See Jones Rpt. at § 8.2.2.12, § 8.2.2.13 (¶¶ 205-214).*

³⁴ *See Jones Rpt. at § 8.2.2.13, ¶ 214.*

As Dr. Goodnight, the CEO of SASI, explained, [REDACTED]

[REDACTED]

[REDACTED]

[REDACTED].³⁶ [REDACTED]

[REDACTED].³⁷ The SAS 76 Manual acknowledges taking from PL/I.³⁸

Output generated by a SAS language program includes unprotectable mathematical and statistical variables and constants, graphs, charts, tables, plots, user data, and calculated values.³⁹

See Eng'g Dynamics, 26 F.3d at 1343. Output is also contributed to or dictated by users and not original to SASI. Also, Dr. Goodnight testified that the choices of what functions to include in the SAS software is often based on user input, suggestions, and vote as opposed to just internal expressive choices.⁴⁰ These elements are not protectable and should be filtered.

Additionally, SAS software has “engines” that access files created with other vendors’ software as if those files were written by SAS software. *See* Ex. M. Similarly, SAS software supports processing of SQL (e.g., in PROC SQL). SQL is not original to SAS. This is not unlike SASI’s complaint that WPS can process programs written in the SAS Language. *See, e.g.*, Dkt.

³⁵ Ex. A (Goodnight Depo. Tr.) at 49:1-5; *id.* at 52:17-53:13.

³⁶ *See* Ex. P (Black Duck code base analysis reports showing presence of non-SAS code).

³⁷ *See* Ex. A (Goodnight Dep. Tr.) at 143:19-144:25. Moreover, Mr. Anthony Barr testified that, when developing the SAS language, he took lexical features from PL/I as well as features that were in SPSS and that he used algorithms and procedures from other textbooks, manuals, and programs. *See* Ex. Q (Barr Dep. Tr.) at 15:13-16:24, 17:21-19:9; 26:3-18, 54:21-55:21.

³⁸ Jones Rpt. Ex. I (SAS 76 Manual) at 1, 37, 300, 319.

³⁹ *See* Jones Rpt. at § 8.2.2.3 (¶¶ 134-38), § 8.2.2.5 (¶¶ 146-50), § 8.2.2.7 (¶¶ 173-79), § 8.2.2.9 (¶¶ 185-91), § 8.2.3 (¶¶ 215-29).

⁴⁰ Ex. A (Goodnight Dep. Tr.) at 65:23-66:24. Dr. Goodnight also went on to testify that, if a user suggested that an output screen look a certain way, and SAS implemented that change, SAS would claim the copyright to that output screen. *Id.* at 73:3-8.

264 at 15. But in any event, these file formats are not original to SASI, and this type of file compatibility is not copyrightable. *See* 46 F.3d at 410 (explaining *Engineering Dynamics* “cannot properly be read to extend...to the practice employed by users of programs of analyzing application programs to ‘read’ the file formats of other programs.”). This must all be filtered.

After filtration, there remains no core protectable expression.⁴¹

VI. THE COURT SHOULD DENY SAS INSTITUTE’S REQUEST FOR DELAY

In closing, SASI—the *plaintiff*—asks the Court to delay its case (despite its pleadings of irreparable injury). This is the third, separate request for a delay that SASI has made unrelated to the pandemic. The Court should reject SASI’s request.

As discussed throughout this brief, the *Oracle* decision has limited applicability to this case, and SASI’s reliance on *Oracle* is flawed. But *Oracle* will either be affirmed in relevant part—in which SASI’s arguments flawed arguments remain flawed in the same way—or it will be reversed—and SASI’s claimed copyrightability in its “input formats” will be conclusively rejected. SASI has also not been diligent. It has known of the October 2020 argument date since at least March 2020. If this case was sufficiently important to warrant delay, SASI could have requested the case to be stayed until after issuance of the *Oracle* decision before the parties fully briefed summary judgment and other pre-trial matters the first time. There is no reason to delay this case simply so SASI can re-brand its flawed copyright theory yet again.

VII. CONCLUSION

WPL respectfully requests that the Court find that SASI failed to establish that, after a valid *Altai* analysis, there remains a core of protectable non-literal elements, and that the Court adopt WPL’s proposed findings of fact and conclusions of law (Dkt. No. 412).

⁴¹ *See* Jones Rpt. at § 8.2.2, § 8.2.3, § 8.4, ¶ 283.

Dated: September 21, 2020

Respectfully submitted,

Bradley W. Caldwell
Bradley W. Caldwell
Texas Bar No. 24040630
Email: bcaldwell@caldwellcc.com
John Austin Curry
Texas Bar No. 24059636
Email: acurry@caldwellcc.com
John F. Summers
Texas State Bar No. 24079417
Email: jsummers@caldwellcc.com
Warren J. McCarty, III
Texas State Bar No. 24107857
Email: wmccarty@caldwellcc.com
Hamad M. Hamad
Texas State Bar No. 24061268
Email: hhamad@caldwellcc.com
CALTWELL CASSADY CURRY P.C.
2121 N. Pearl St., Suite 1200
Dallas, Texas 75201
Telephone: (214) 888-4848
Facsimile: (214) 888-4849

ATTORNEYS FOR DEFENDANTS
WORLD PROGRAMMING LIMITED,

Charles Everingham IV
State Bar No. 00787447
ce@wsfirm.com
T. John “Johnny” Ward Jr.
Texas Bar No. 00794818
jw@wsfirm.com
Claire Henry
Texas Bar No. 24053063
claire@wsfirm.com
Andrea L. Fair
Texas Bar No. 24078488
andrea@wsfirm.com
WARD, SMITH & HILL, PLLC
P.O. Box 1231
Longview, TX 75606
903-757-6400
903-757-2323 (Facsimile)

ATTORNEYS FOR DEFENDANTS
WORLD PROGRAMMING LIMITED

CERTIFICATE OF SERVICE

I certify that on September 21, 2020, this document was served on counsel of record by email.

/s/ Hamad M. Hamad
Hamad M. Hamad

CERTIFICATE OF AUTHORIZATION TO FILE UNDER SEAL

I certify that this document is authorized to be filed under seal pursuant to this case's Protective Order.

/s/ Hamad M. Hamad
Hamad M. Hamad